Organisme(s) : EDF-R&D/AMA

Manuel d'Utilisation
Fascicule U1.0- : Introduction à *Code_Aster*
Document : U1.04.00

Interface d'accès à *Code_Aster* : astk

Résumé :

Ce document présente astk (contraction de *Aster* et Tk, prononcer « astek »), l'interface permettant d'organiser ses calculs *Aster*.
On décrit la logique de fonctionnement d'astk, l'interface graphique elle-même, bsf qui permet d'explorer les fichiers sur les différents serveurs, puis on explique comment l'utilisateur peut configurer astk en fonction de ses besoins.
Enfin, sur un exemple concret, on montre comment utiliser astk pour faire une étude, une surcharge, lancer une liste de cas-tests.
Pour les personnes ayant accès au serveur de calcul *Aster* du réseau EDF, on explique comment accéder aux fiches de retour d'expérience (REX), et, pour les développeurs, aux outils de l'AGLA (atelier de génie logiciel *Aster*).

Sont décrites ici les fonctionnalités d'astk version 1.2 qui supporte les versions 6 (à partir de la STA6.5), 7 et 8 de *Code_Aster*.

La version 5 n'est pas supportée par astk.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Table des matières

1 Présentation et notions générales....4
1.1 Démarrage....4
1.2 Fonctionnalités....4
1.3 Mode de fonctionnement....5
1.4 Serveurs astk....5
1.5 Profil astk....6
2 Description de l'interface graphique....7
2.1 Barre de menu....7
2.1.1 Menu Fichier....7
2.1.2 Menu Configuration....8
2.1.3 Menu Outils....9
2.1.4 Menu Options....9
2.1.5 Menu Aide....10
2.2 Les onglets....10
2.2.1 Onglets ETUDE, TESTS, SURCHARGE....10
2.2.1.1 Liste des types pour ETUDE....12
2.2.1.2 Liste des types pour TESTS....12
2.2.1.3 Liste des types pour SURCHARGE, partie DONNEES....12
2.2.1.4 Liste des types pour SURCHARGE, partie RESULTATS....13
2.2.2 Bouton AGLA....13
2.2.3 Bouton REX....14
2.3 Paramètres de soumission....18
2.4 Barre d'état....19
2.5 Suivi des jobs....19
3 Boite de sélection de fichiers : bsf....20
3.1 Navigation....21
3.2 Menu Fichier et menu contextuel....22
3.3 Menu Affichage....22
4 Configuration....23
4.1 Serveurs....24
4.2 Interface....25
4.3 Outils....26
4.4 Impression....26
5 Comment faire une étude ?....27
5.1 Création du profil....27
5.2 Sélection des fichiers....27
5.2.1 Définition d'un chemin de base....27
5.2.2 Ajout de fichiers existants....27

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5.2.3 Ajout de fichiers….27
5.2.3.1 …en insérant une ligne vide.28
5.2.3.2 …avec « Valeur par défaut ».28
5.2.4 Supprimer un fichier.28
5.3 Lancement du calcul.29
5.4 Consultation des résultats.29
5.5 Utilisation des outils.29
6 Comment réaliser une surcharge ?.30
6.1 Ajout des sources.30
6.2 Définir les résultats de la surcharge.30
6.3 Prise en compte de la surcharge.30
6.3.1 Préparer la surcharge seule, puis lancer une étude.30
6.3.2 Faire la surcharge et lancer l'étude en même temps.31
7 Comment lancer une liste de tests ?.31
8 Questions fréquentes.32
Annexe 1 Utilisation du service as_run (de ASTK_SERV).33

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1 Présentation et notions générales

La mise à disposition d'un outil comme *Code_Aster* qui fonctionne sur de multiples plates-formes nécessite une interface graphique pour simplifier le travail de l'utilisateur.
Par ailleurs, l'évolution des moyens de calcul centralisés et la diffusion en libre de *Code_Aster* ont nécessité le développement d'un produit portable dont l'architecture puisse s'adapter aux différentes configurations informatiques.

astk est l'interface graphique qui permet d'organiser ses calculs *Aster* : préparer ses données, organiser les fichiers, accéder aux outils de pré et post-traitement, lancer et suivre l'évolution des calculs. astk permet également de choisir la version de *Code_Aster* à utiliser parmi celles disponibles (STA, NEW…). Seules les versions 6 et suivantes sont supportées par astk.

Le nom « astk », prononcer « astek », vient de la contraction de *Aster* et Tk qui est la librairie graphique utilisée (comme asterix était issu de *Aster* et X pour X Windows).

1.1 Démarrage

Pour lancer l'interface, il suffit de taper dans un terminal : `astk`

En général, la commande a été placée dans le chemin par défaut des utilisateurs, l'interface s'ouvre avec une configuration par défaut (voir [*§ 4] Configuration*).
Si la commande n'est pas trouvée, il faut lancer : `[install_dir]/ASTK_CLIENT/bin/astk` où `[install_dir]` est le répertoire d'installation d'astk (contactez éventuellement votre administrateur).

Accès au serveur de calcul *Aster* (EDF et prestataires)

Sur le serveur de calcul Aster d'EDF, utiliser la commande `/aster/outils/astk` *.*
Il est conseillé aux unités EDF d'installer l'interface sur un serveur départemental.

1.2 Fonctionnalités

ETUDE : astk permet de lancer un calcul *Aster* sur la machine locale (par exemple dans le cadre d'une utilisation sur ordinateur personnel), sur un serveur de calcul départemental ou bien sur le serveur de calcul *Aster* de EDF-R&D (accès restreint à EDF et ses prestataires d'études).

SURCHARGE : La diffusion du code source de *Code_Aster* autorise chacun à tester ses propres développements. astk permet de "surcharger" le code, c'est-à-dire ajouter ou modifier des fichiers sources, de créer une version particulière et l'utiliser sur des cas-tests ou pour des études. On peut ainsi créer et utiliser de nouveaux exécutables, catalogues de commandes ou d'éléments, et modules python.

TESTS : Il est souvent judicieux de tester que ses propres développements n'impactent pas le code par ailleurs ; astk permet de lancer facilement une liste de cas-tests avec une version personnelle.

AGLA : Aux développeurs de la version de référence EDF, astk offre l'accès à l'atelier de génie logiciel *Aster*, AGLA, qui permet d'organiser le développement collaboratif de la version de développement (éviter les conflits, assurer la non régression…).

REX : Faire vivre le code passe par la prise en compte des remarques, besoins, avis des utilisateurs. Des fiches de retour d'expérience peuvent être émises et consultées par les utilisateurs depuis l'interface, les développeurs peuvent y répondre (accès au serveur EDF requis).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

MULTI-MACHINES : Les fichiers nécessaires à ses différentes actions (fichiers de maillage, source, résultats…) peuvent être répartis sur différentes machines sur le réseau (déclarées dans astk), astk assurant le transfert et la compression/décompression.
Le passage d'un coupe-feu n'est pas proposé.

OUTILS : L'utilisateur peut lancer différents outils pré-définis et configurer ceux dont il a besoin (mailleur, outil de post-traitement, éditeur…).

BSF : Un navigateur est fourni (appelé `bsf`, boîte de sélection de fichiers), il permet de parcourir les systèmes de fichiers des machines distantes définies et d'effectuer des opérations courantes sur les fichiers : copie, suppression, changement de nom, impression, ou encore d'ouvrir une fenêtre de commandes sur ces machines.

ASJOB : On peut suivre les calculs, leur état (notamment dans le cas de lancement en batch) depuis la fenêtre « Suivi des jobs », aussi appelée `asjob`.

1.3 Mode de fonctionnement

L'architecture client/serveur autorise une séparation nette entre l'interface (client) et les outils utilisés pour accéder au code (services). Elle permet d'utiliser l'ensemble des scripts de l'AGLA pour gérer la version de référence en batch et `as_run` (cf. Annexe 1), un service astk, pour les autres versions disponibles en local ou sur le réseau.
Le protocole de communication entre les différentes machines est rsh ou ssh pour les commandes shell et rcp, scp ou rsync pour la copie de fichiers.

Exemple : Le client (c'est-à-dire l'interface lancée par la commande `astk`) demande le lancement d'un calcul sur un serveur de calcul.
L'interface exécute un service sur le serveur en question (commande rsh ou ssh) en lui indiquant les paramètres nécessaires au calcul (contenu dans un profil, voir [§ 1.5]). Le service prend ensuite en charge les opérations nécessaires : rapatrier les fichiers nécessaires au calcul des différentes machines vers le serveur de calcul, lancement du calcul, retour des fichiers résultats…

1.4 Serveurs astk

Pour astk, un serveur est :

- soit un serveur de calcul *Aster* : une machine sur laquelle on peut trouver la partie "serveur" de astk, c'est-à-dire l'ensemble des services qui permettent d'accéder au code ; on pourra lancer des calculs via ce serveur et utiliser des fichiers sur ce serveur pour un calcul,
- soit un serveur de fichiers : on pourra simplement utiliser des fichiers sur ce serveur lors d'un calcul.

Le serveur appelé "Local" est en fait un serveur de fichiers (seules les informations de connexion sont nécessaires pour un serveur de fichiers, or sur la machine locale, on les connaît facilement).
Si l'on souhaite lancer des calculs sur la machine locale (sur laquelle sont installés une version d'*Aster* et la partie serveur d'astk), il faut aussi déclarer cette machine comme un serveur de calcul (Local et Machine dans l'exemple suivant).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Exemple de configuration : (voir [§ 2.1.2])

astk est utilisé sur la machine de nom `mach00`, sur laquelle est installé *Code_Aster*. On a accès à un serveur de fichiers départemental `file01`, les fichiers de ce serveur sont accessibles (montage nfs) depuis deux machines de calcul `comp02`, `comp03`. On a aussi accès à un cluster avec N nœuds de calcul dont la machine frontale est `front04`.

On a alors :

- « Local » (label réservé) : on l'a toujours pour explorer les fichiers qui se trouvent sur `mach00` ;
- « Machine » (label quelconque, la procédure d'installation fixe ce nom à la valeur retournée par la commande `` `uname -n` ``) : dont l'adresse IP est celle de `mach00` qui est le serveur de calcul (services astk installés) de la machine locale ;
- « Depart » (label quelconque) : dont l'adresse IP est celle de `file01`, qui permet d'explorer les fichiers hébergés par le serveur de fichier, configuré comme un serveur de calcul Aster (services astk installés) ayant `comp02` et `comp03` vus comme des nœuds de calcul (`comp0i` peut être la même machine que `file01`) ;
- « Cluster » (label quelconque) : dont l'adresse IP est celle de `front04` qui est un serveur de calcul (services astk installés) ayant N nœuds, la seule machine accessibles étant `front04`.

La différence entre « Depart » et « Cluster » est qu'en général les stations de calcul sont accessibles directement (on pourrait donc soumettre un calcul en interactif sur l'une ou l'autre), alors que pour un cluster, les utilisateurs ne voient en général que la machine frontale, la répartition étant faite sur les nœuds par un séquenceur de travaux batch.

On peut aussi bien avoir plusieurs serveurs de calcul « Depart » ou « Cluster » que n'avoir uniquement le poste « Local+Machine ».

1.5 Profil astk

Un profil astk est un fichier qui contient toutes les informations relatives à votre étude, votre surcharge, etc. : l'emplacement des fichiers de l'étude, en donnée, en résultat, le type associé à chaque fichier, les paramètres de soumission du calcul (mémoire, temps, machine de calcul, batch/interactif…), en cas de surcharge, l'emplacement des fichiers sources, de l'exécutable, des catalogues produits…

Le profil contient également des paramètres sur l'interface elle-même pour reprendre une étude exactement dans le même état qu'on l'avait laissée.
Le profil est enregistré sur demande de l'utilisateur (menu *Fichier/Enregistrer* ou *Enregistrer sous…*) et à chaque lancement d'un calcul.

Pour les utilisateurs d'asterix : le profil astk englobe le profil d'étude et d'exécution au sens asterix, et les paramètres de soumission qui n'étaient pas stockés dans asterix. Le fichier, même s'il reste au format ASCII, est malgré tout plus compliqué à lire.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2 Description de l'interface graphique

L'interface graphique (IHM) se décompose en 4 parties :

1) Une barre de menu
2) Les onglets
3) Les paramètres de soumissions
4) La barre d'état

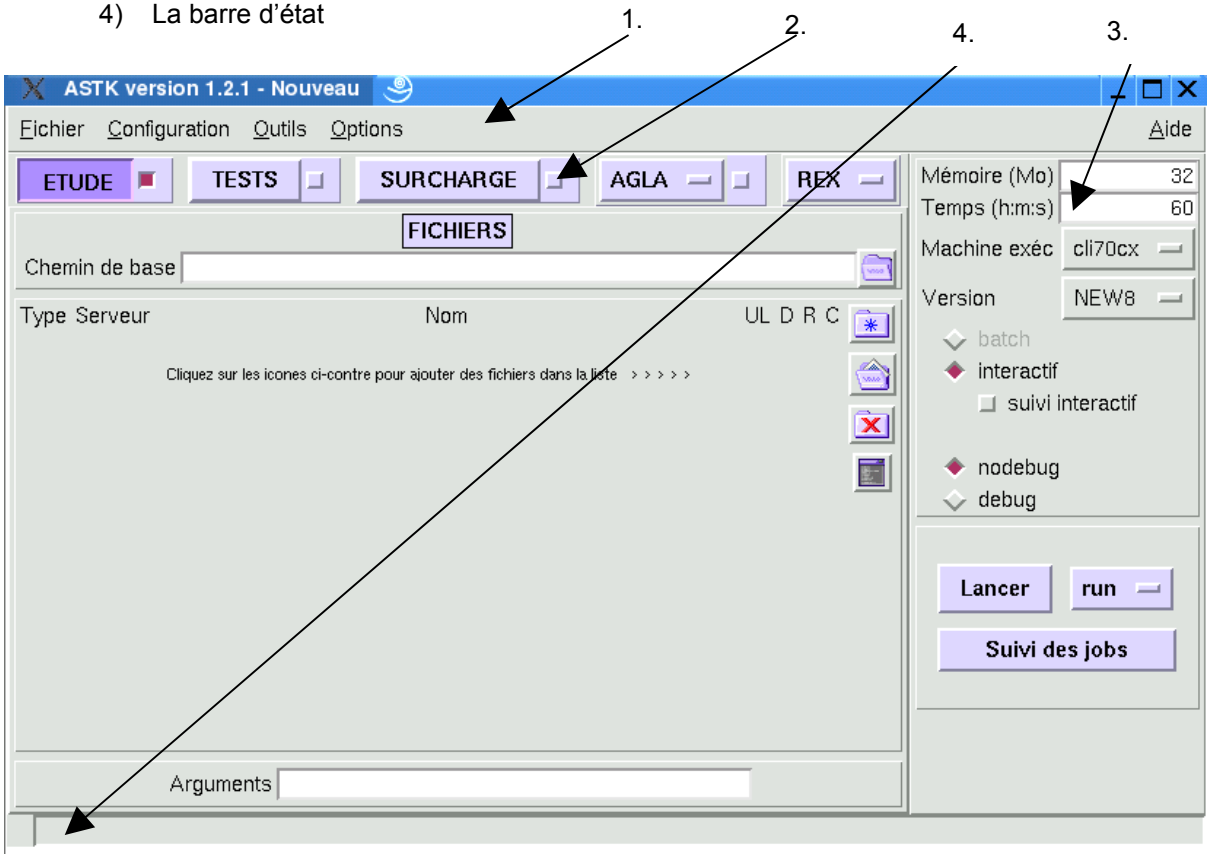


Figure 2-a : Fenêtre principale

2.1 Barre de menu

Le menu « Aide » permet également d'accéder à la description des menus.

2.1.1 Menu Fichier

- Nouveau : Permet de créer un nouveau profil.
- Ouvrir : Ouvre un profil créé auparavant par `astk`. On ne peut pas ouvrir les profils créés par asterix de cette façon.
- Enregistrer : Sauvegarde le profil courant (ceci est fait automatiquement lors du lancement d'un calcul).
- Enregistrer sous : Sauvegarde le profil courant en donnant le choix du nom du profil et de son emplacement.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- Copier sous : Copie le profil courant dans un autre répertoire et propose de copier les fichiers (en données et/ou résultats) qu'il référence.

- Importer :
  - asterix : Relit un fichier .pret ou .prex produit par asterix (le contenu du .pret étant moins riche, des retouches seront probablement nécessaires, consulter les INFO).
  - astk_serv : Relit un fichier .export d'une exécution précédente (ou le fichier `*.pNNN` du répertoire `$HOME/flasheur`).

- Exporter sous : Permet d'exporter le profil courant au format astk_serv (pour les services ASTK_SERV).

- Impression : Configuration des commandes d'impression
  L'utilisateur définit des commandes d'impression (lpr, a2ps...) qui seront utilisées pour imprimer les fichiers, et les files d'impression (imprimantes). La commande peut être exécutée sur un serveur distant.
  Au code @P de la ligne de commande sera substitué le nom de la file d'impression, le nom du fichier remplacera @F (voir [§ 2.1.3] pour les codes de champs).
  Exemple de commande d'impression :
  ```
  lpr -h -P @P @F
  ```
  ou
  ```
  /usr/bin/a2ps -P @P -2 --print-anyway=yes --sides=tumble @F
  ```

- Quitter : Termine la session

- Les N derniers profils ouverts sont directement accessibles à partir du menu Fichier.

2.1.2 Menu Configuration

- Serveurs : Permet de modifier la configuration des serveurs : serveurs de calcul ou serveurs de fichier.
  - Pour les serveurs de calcul, choisir le mode de téléchargement de la configuration *Aster* : le serveur retourne la liste des versions disponibles, le mode d'exécution (batch et/ou interactif) et les limites associées.
    Il faut indiquer le 'login' autorisé à se connecter au serveur, et le répertoire où est installée la partie serveur de `astk` (ASTK_SERV).
  - Pour les serveurs de fichier, choisir "aucun" comme mode de téléchargement de la configuration Aster.

- Interface : Définit les préférences de l'utilisateur :
  Pour `astk` :
  Nom, prénom, email seront utilisés pour le lien vers l'outil de gestion du retour d'expérience (émission, consultation des fiches anomalies, d'évolution...).
  L'accès aux fonctions de l'atelier de génie logiciel *Aster* si l'utilisateur est reconnu en tant que développeur Aster sur la machine de référence.
  Le chemin d'accès à l'éditeur et au terminal (fenêtre xterm) sur la machine locale, la version à sélectionner par défaut, le nombre de profils rémanents dans le menu Fichier, le niveau de message (debug), le nom de domaine réseau de la machine et si le mode DHCP (adresse IP dynamique) est actif, les protocoles de communication utilisés pour les commandes shell (rsh ou ssh) et la copie de fichiers (rcp, scp ou rsync).
  Pour le suivi des jobs:
  Le nombre de lignes affichées lors de la consultation d'un calcul en cours d'exécution (tail), et la fréquence d'actualisation automatique en minutes.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.1.3 Menu Outils

- Paramètres : Permet de configurer le lancement des outils.
  Certains outils sont prévus en standard (sans forcément être disponibles sur la machine locale) :
  - `bsf` : Explorateur de fichiers multi-machines,
  - Eficas : Editeur de fichier de commandes *Aster*,
  - Gibi : Mailleur et outil de post-traitement (utilisable gratuitement avec *Aster*),
  - GMSH : Mailleur et outil de post-traitement (libre),
  - Stanley : Outil de post-traitement intégré à *Aster*,
  - Foresys : Outil de navigation dans les sources fortran.

  Lorsque l'on choisit de lancer Stanley, astk cherche dans le profil les bases disponibles (par ordre de préférence une `base` en résultat, s'il n'y en a pas, une `bhdf` (base HDF) en résultat, sinon une `base` en donnée et finalement, une `bhdf` en donnée), produit un profil temporaire à partir du profil courant avec un fichier de commandes Aster qui commence par `POURSUITE()` et qui lance `STANLEY()`.

  L'utilisateur peut ajouter ses propres outils, paramétrer le chemin d'accès aux outils (y compris modifier la commande d'accès aux outils standards), définir à quels types de fichiers associer l'outil et préciser si on peut utiliser l'outil sur un fichier distant.

  Les codes suivants peuvent être utilisés dans la ligne de commande :
  - @F : chemin absolu du fichier sélectionné,
  - @R : répertoire contenant le fichier sélectionné,
  - @f : nom du fichier (sans le répertoire),
  - @D : adresse du DISPLAY (celui connu au moment du lancement de l'interface).

  Les outils sont appelés soit à partir du menu Outils, soit à partir du menu contextuel sur un fichier d'une liste ou dans l'explorateur (bouton droit).
  Pour sélectionner un fichier, il suffit de cliquer sur son nom dans une liste (dans l'onglet Etude, Tests ou Surcharge).
  En double-cliquant sur un fichier, l'association entre le type du fichier et l'outil à utiliser s'appuie sur l'extension du nom de fichier pour la bsf, alors que le type sélectionné par la liste déroulante (cf. description des onglets) prévaut dans astk.

2.1.4 Menu Options

- Arguments :
  Permet de positionner des arguments optionnels qui seront passés sur la ligne de commande de l'exécution de *Code_Aster*.
  `dbgjeveux` : active un mode différent de la gestion des objets en mémoire pour détecter les écrasements et destruction d'objets,
  `rep_outils` : définit le répertoire où sont cherchés les outils appelés depuis *Aster* (onmetis ou gibi par exemple),
  `rep_dex` : définit le répertoire où sont cherchées les données externes (données de maillage),
  `rep_mat` : définit le répertoire où sont stockées les données du catalogue matériau.

- Paramètres :
  Définition des paramètres optionnels utilisés lors du lancement d'une liste de tests (bouton TESTS).
  `nbmaxnook` : nombre maximal de cas-tests invalides (NOOK, ARRET_ANORMAL...) avant l'arrêt de la liste des tests,
  `cpresok` : permet de choisir quels sont les tests dont on garde les fichiers résultat,
  RESOK : on garde les fichiers des tests OK et NOOK,
  RESNOOK : on ne garde que les fichiers des tests NOOK,
  `facmtps` : facteur multiplicatif du temps des tests (par rapport au temps de référence du .para). Utile, par exemple, quand on soumet des tests avec un exécutable construit en mode debug, plus lent.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.1.5 Menu Aide

- Introduction/Paramètres du calcul/Menus : accès au texte d'aide
- Historique des modifications : Evolution d'astk au fil des versions, nouvelles possibilités, corrections effectuées, anomalies connues...
- Problèmes fréquents : Quelques questions fréquemment posées avec leurs solutions.
- Fenêtre des messages : Les messages d'informations <INFO>, d'erreurs <ERREUR> sont écrits dans cette fenêtre. Avec un niveau de message supérieur ou égal à 1 (voir [§ 2.1.2]), on obtient plus ou moins d'informations.
  Le niveau 3 affiche les commandes soumises aux serveurs, les niveaux 4 et 5 permettent d'avoir des détails complémentaires dans le fichier `.o` (de type debug) lors de l'exécution de `as_run` (en interactif) ; ce qui permet de comprendre parfois pourquoi une erreur se produit.
- A propos : l'incontournable fenêtre d'information.

2.2 Les onglets

On trouve cinq boutons dans cette zone. On parle d'onglet parce que le contenu que l'on trouve sous les boutons change selon celui qui est pressé. Graphiquement, ce ne sont pas de « vrais » onglets car le widget n'existe pas dans la librairie Tk standard (et on n'a pas voulu ajouter un pré-requis supplémentaire !).

ETUDE, TESTS et SURCHARGE sont des onglets dans lesquels on renseigne la liste des fichiers nécessaires ; AGLA et REX sont des boutons qui peuvent utiliser le contenu des autres onglets.

Enfin, la case à cocher située à côté des quatre premiers boutons signale que l'on utilise (ou non) le contenu de l'onglet associé.
Exemples : Pour lancer une étude, il faut cocher l'onglet ETUDE. Pour faire appel aux fonctions de l'AGLA (sur la machine de référence), il faut cocher l'onglet AGLA ; on notera que dans ce cas, TESTS et SURCHARGE sont automatiquement cochés car leur contenu est pris en compte.

2.2.1 Onglets ETUDE, TESTS, SURCHARGE

Pour accéder plus facilement aux fichiers, parcourir les arborescences plus rapidement (les fichiers étant souvent regroupés dans des répertoires proches), ou encore simplifier l'affichage des noms de fichiers, on peut définir un **chemin de base**.

Figure 2.2.1-a : Chemin de base

On le définit en cliquant sur le bouton Parcourir et on choisit le répertoire qui sera le répertoire par défaut.

Chaque onglet contient une liste de fichiers (deux listes pour SURCHARGE).

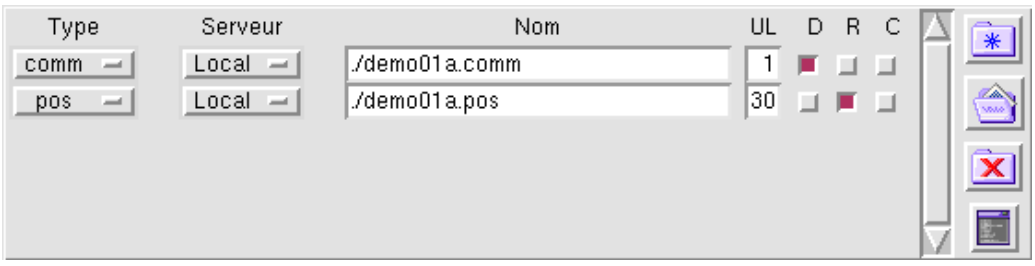


Figure 2.2.1-b : Liste de fichiers/répertoires

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

On définit un fichier ou répertoire par ligne. On trouve de la gauche vers la droite :

- Type : type du fichier ou du répertoire ;
- Serveur : nom du serveur sur lequel se trouve le fichier/répertoire ;
- Nom : chemin d'accès au fichier : en absolu s'il commence par '/', en relatif par rapport au chemin de base dans le cas contraire ;
- UL : numéro d'unité logique associé à ce fichier ;
- D, R, C : cases à cocher pour indiquer si le fichier/répertoire est respectivement en donnée, résultat, compressé (avec gzip).

Lorsque l'on change le type d'un fichier/répertoire, des valeurs par défaut sont positionnées pour les indicateurs D, R, C et le numéro d'unité logique.

Remarque sur les types :

> *Code_Aster manipule les fichiers via les procédures fortran qui utilisent des numéros d'unité logique (fichier `fort.19` par exemple) ; il affecte donc des numéros d'unité logique par défaut pour simplifier la vie de l'utilisateur. Le « type » permet ainsi d'accéder facilement aux numéros utilisés par défaut ; il permet aussi à astk de vérifier que l'utilisateur fournit des données cohérentes (par exemple que l'on fournit bien un répertoire pour tel type, ou que l'on ne fournit pas deux objets alors qu'un seul est autorisé pour tel autre…).*

Pour manipuler la liste, on dispose de quatre boutons :

- : permet d'ajouter une nouvelle entrée vierge à la fin de la liste ;
- : permet d'ajouter un fichier/répertoire à la fin de la liste en parcourant l'arborescence de fichiers ;
- : supprimer la ligne actuellement sélectionnée dans la liste ;
- : ouvre le fichier/répertoire actuellement sélectionné dans la liste avec l'éditeur du serveur où se trouve le fichier (cf. [§ 4.1]). S'il s'agit d'un répertoire, tous les fichiers du répertoire sont ouverts avec l'éditeur (attention aux répertoires contenant beaucoup de fichiers ou des fichiers binaires non éditables !).

Arguments (onglet ETUDE et TESTS seulement) :
Cette zone de texte permet de transmettre des arguments à l'exécutable *Aster*. Voir aussi [§2.1.4].

Menu contextuel :
En cliquant avec le bouton droit sur une entrée de la liste, on accède à un menu contextuel :

- Ouvrir : lance l'outil associé à ce type de fichier (déterminé par le champ « Type », et non l'extension du fichier), si aucun outil n'est associé à ce type, on édite le fichier ;
- Editer : édite le fichier (ou tous les fichiers du répertoire) de la même manière que le bouton ;
- Imprimer : imprime le fichier sélectionné ;
- Ouvrir avec… : on peut choisir d'ouvrir le fichier sélectionné avec un des outils disponibles (le résultat peut être étonnant si l'outil ne connaît pas ce type de fichier !) ;
- Valeurs par défaut : l'interface détermine un nom de fichier par défaut en fonction du « Type » choisi à partir du nom du profil (fichier `.astk`), aucune valeur n'est proposée si le profil n'a pas encore été enregistré (s'utilise en général sur une ligne vierge que l'on vient d'insérer, les indicateurs D/R/C ne sont pas affectés par cette opération) ;
- Propriétés : affiche les permissions, la taille, la date et l'heure du fichier (commande `ls -la`).

Onglet SURCHARGE

> *Les données (fichiers sources) sont fournies dans la liste supérieure, alors que les résultats (exécutable, catalogues compilés) sont fournis dans la liste inférieure.*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.2.1.1 Liste des types pour ETUDE

comm :	fichiers de commande *Aster* (y compris les fichiers de poursuite)
mail :	fichier maillage au format *Aster*
erre :	fichier d'erreur (fort.9 d'Aster)
mess :	fichier des messages de l'exécution
resu :	fichier de résultat (impression des tests, impression au format *Aster*)
base :	répertoire contenant la base du calcul
bhdf	répertoire contenant la base du calcul au format HDF
cast :	fichier résultat au format CASTEM
mast :	gardé pour raison compatibilité
mgib :	maillage au format Gibi
mmed :	maillage au format MED
msh :	maillage au format Gmsh
msup :	maillage au format IDEAS
pos :	fichier résultat au format Gmsh
ensi :	répertoire résultat au format Ensight
dat :	fichier résultat contenant des courbes au format XMGRACE
ps :	Fichier postscript
agraf :	fichier résultat contenant les données pour Agraf (les anciennes versions d'Aster écrivait les directives et les données dans un même fichier qu'il fallait découper avec la commande `post_agraf` sur la machine de référence)
digr :	fichier résultat contenant les directives pour Agraf
rmed :	fichier résultat au format MED
unv :	fichier résultat au format « UNiVersel » (IDEAS)
para :	fichier de paramètres (retranscription des paramètres du calcul pour les tests)
repe :	répertoire en données et/ou résultats (permet de transmettre/récupérer le contenu complet d'un répertoire ; comme on n'affecte pas de numéros d'unité logique *Aster* doit accéder aux fichiers par leurs noms, cf. commande `DEFI_FICHIER`)
libr :	fichier ou répertoire au choix de l'utilisateur
btc :	script de lancement généré par le service (on peut ainsi le récupérer, le modifier…).

Lors d'un astout sur la machine de référence, `resu_test` doit être sur celle-ci.

2.2.1.2 Liste des types pour TESTS

list :	fichier contenant la liste des tests à exécuter (un nom de test par ligne sans `.comm`)
rep_test :	répertoire contenant les fichiers de données des tests (commandes, maillage…)
resu_test :	répertoire où sont recopiés les fichiers résultats
btc :	idem ETUDE

2.2.1.3 Liste des types pour SURCHARGE, partie DONNEES

f :	sources fortran
c :	sources C
py :	sources Python
capy :	sources des catalogues de commandes
cata :	sources des catalogues d'éléments, d'options, de grandeurs…
hist :	fichier histor (historique des modifications)
conf :	fichier de configuration (choix des options de compilation, des librairies…)
unig :	fichier `unigest` contenant les modules à supprimer
datg :	données géométriques
cmat :	catalogues de données matériaux

En général, on fournit un répertoire pour les cinq premiers types (sources) ; tous les fichiers dont l'extension correspond au type indiqué sont pris en compte. Néanmoins, il est déconseillé de mélanger les types de fichiers dans un même répertoire.
hist, unig, datg et cmat sont exclusivement liés à la gestion de la version de référence.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.2.1.4 Liste des types pour SURCHARGE, partie RESULTATS

exec : fichier de l'exécutable *Aster*
cmde : répertoire du catalogue de commande compilé
ele : fichier du catalogue d'éléments
forlib : librairie pour FORESYS
btc : idem ETUDE

forlib n'est utilisé que sur la machine de référence.

2.2.2 Bouton AGLA

Ce bouton permet d'accéder aux fonctions de l'atelier de génie logiciel qui coordonne les actions des développeurs de la version de cohérence. Pour cela, la case à droite du bouton doit être cochée, ce qui a pour effet de prendre en compte automatiquement le contenu des onglets TESTS et SURCHARGE, il suffit ensuite de cliquer sur le bouton « Lancer ».
On se reportera au manuel de l'AGLA ([D1.02.01]) pour plus de détails sur les différentes actions.

- **ASNO** : permet de « noter » des modules (signaler que l'on prévoit de restituer une modification), s'applique aux fichiers sources et aux fichiers de test.
- **ASDENO** : permet de « dénoter » des modules. Cette action n'utilise aucune des données de TESTS ou SURCHARGE. Lorsque l'on clique sur « Lancer », une fenêtre demande de choisir le type de module à dénoter (fortran/C, python, catalogues ou test), et d'indiquer le nom des modules séparés par un espace, une tabulation ou un retour à la ligne (**sans l'extension** : `op0191` pour dénoter le fichier `op0191.f`). Pour les fichiers Python, il est nécessaire de préciser le nom du répertoire (package dans la dénomination Python) dans lequel ils se trouvent (car contrairement aux sources fortran et C, le même nom peut être utilisé dans des répertoires différents), exemple : `macr_recal_ops@Macro` (désigne le module `macr_recal_ops` du package `Macro`).

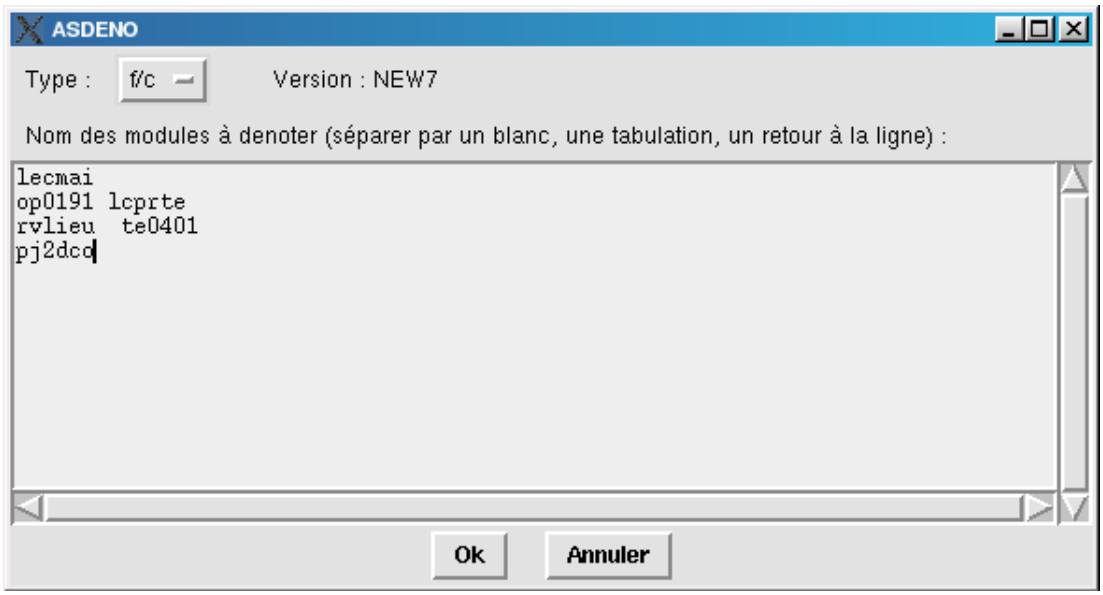


Figure 2.2.2-a : Fenêtre ASDENO

- **ASQUIT** : fournit dans le fichier output la liste de tous les modules notés.
- **ASVERIF** : vérifie qu'un ensemble de sources peut être intégré dans la version de référence (respect des règles de programmation, fichier histor présent…).
- **PRE_EDA** : il doit être effectué avant la réunion de l'équipe de développement, le développeur signale ainsi son intention de restituer des sources, PRE_EDA soulève les points qui devront être réglés en réunion de développement (PRE_EDA effectue un ASVERIF et un passage des tests fournis dans le profil).

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

- **ASREST** : il s'agit de la phase ultime de la restitution qui consiste en un ASVERIF, un passage des tests de la liste de non régression, le code retour doit être inférieure ou égal à 2 pour que la restitution soit prise en compte.
- **FORLIB création/suppression** : permet de produire/supprimer une librairie pour l'outil FORESYS (sur la machine de référence).

2.2.3 Bouton REX

Remarque

Cette partie sera profondément modifiée courant 2005 avec le changement de l'outil de gestion du retour d'expérience.

Ce bouton permet d'accéder à l'outil de gestion de retour d'expérience (sur la machine de référence). Il est désactivé lorsque la machine de référence n'est pas dans la liste des serveurs.

- **Choix du correspondant** : L'utilisateur doit choisir son correspondant *Aster* parmi ceux qui lui sont proposés. Le correspondant *Aster* valide les fiches émises par ses utilisateurs avant qu'elles soient examinées par l'équipe de développement (il filtre ainsi les problèmes déjà connus ou résolus…). Les développeurs sont leur propre correspondant, pour eux, ce choix du menu est donc désactivé (voir aussi [§ 2.1.2]).
- **Emettre une fiche sans joindre le profil** : Cette fonction permet à l'utilisateur d'émettre une fiche de retour d'expérience pour signaler une anomalie dans *Code_Aster* (AL : anomalie logiciel), demander une évolution de *Code_Aster* (EL : évolution logiciel), dans un des outils associés (Metis, Homard, Eficas, astk, bsf…) (AO : anomalie outil/EO : évolution outil), une modification de la documentation (ED : évolution documentation), ou une demande d'expertise en modélisation (AOM : aide à l'optimisation de la modélisation).
- Les informations relatives à l'émetteur de la fiche sont accessibles par le menu *Configuration/Interface*. Les fichiers contenus dans le profil courant ne sont pas joints à la fiche, ce qui peut rendre très difficile le traitement de la fiche. Il est conseillé d'utiliser "émettre une fiche et y associer les fichiers".
- La version de *Code_Aster* indiquée est celle sélectionnée dans l'interface (*Paramètres du calcul*).
- **Emettre une fiche et y associer les fichiers** : Cette fonction permet d'émettre le même type de fiche que la précédente, cette fois-ci les données incluses dans le profil sont jointes à la fiche. Pour une anomalie, les fichiers pour reproduire l'erreur devraient toujours être fournis.
- Ce qui n'empêche pas l'utilisateur d'essayer d'isoler autant que possible le problème rencontré, notamment d'essayer de joindre une étude sur un modèle qui nécessite peu de mémoire et de temps de calcul !
- **Consulter les fiches** : La fenêtre de consultation des fiches comporte une ligne où l'on définit des critères de recherche, le filtre permet d'effectuer une recherche dans le texte des fiches (peut être assez long) en utilisant une expression régulière.
- « Rechercher » : rafraîchit la liste des fiches ;
- « Visualiser » : ouvre la ou les fiches sélectionnées dans la liste (on peut aussi double-cliquer sur une fiche pour la visualiser) ;
- « Imprimer » : imprime le contenu de la ou des fiches sélectionnées dans la liste ;
- « Fichiers associés » : si la fiche sélectionnée a des fichiers associés, ceci ouvre une bsf où se trouvent ces fichiers ;
- « Etude associée » : si la fiche sélectionnée a des fichiers associés, ceci permet d'importer directement le profil associée à la fiche ;
- « Imprimer la liste » : imprime la liste des fiches affichées.
- **Fiches à corriger** : *Pour les développeurs seulement.* On retrouve la même fenêtre de consultation des fiches avec des critères pré-définis qui correspondent aux fiches que le développeur doit traiter. Un bouton supplémentaire « Réponse définitive » lui permet de répondre à la fiche (après l'avoir soumise à l'équipe de développement).
- **Supprimer une fiche** : Permet de supprimer une fiche qui est encore dans l'état `EMIS_UA`/`EMIS_CUA` par celui qui l'a émise.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

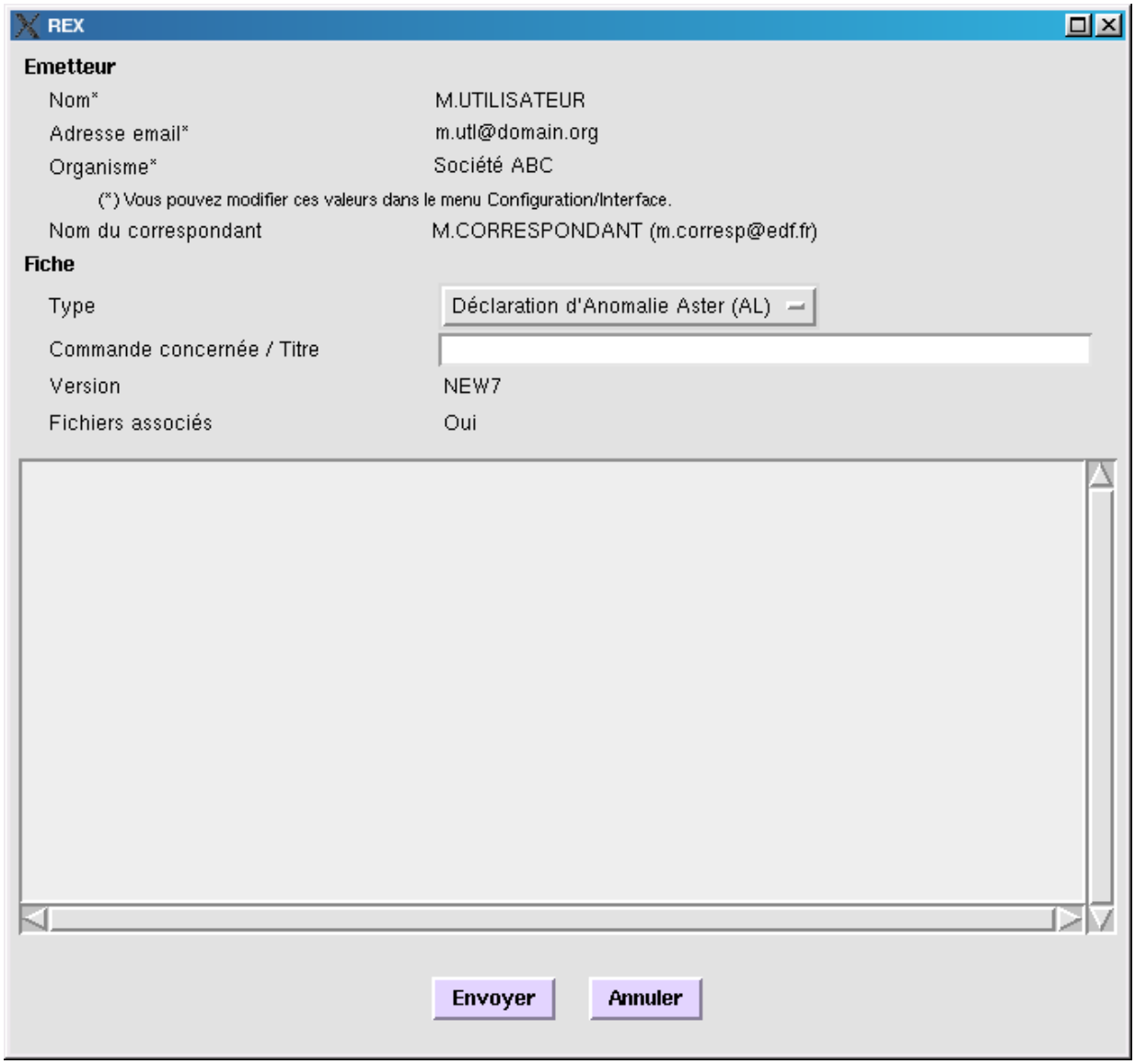


Figure 2.2.3-a : Emission d'une fiche REX

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

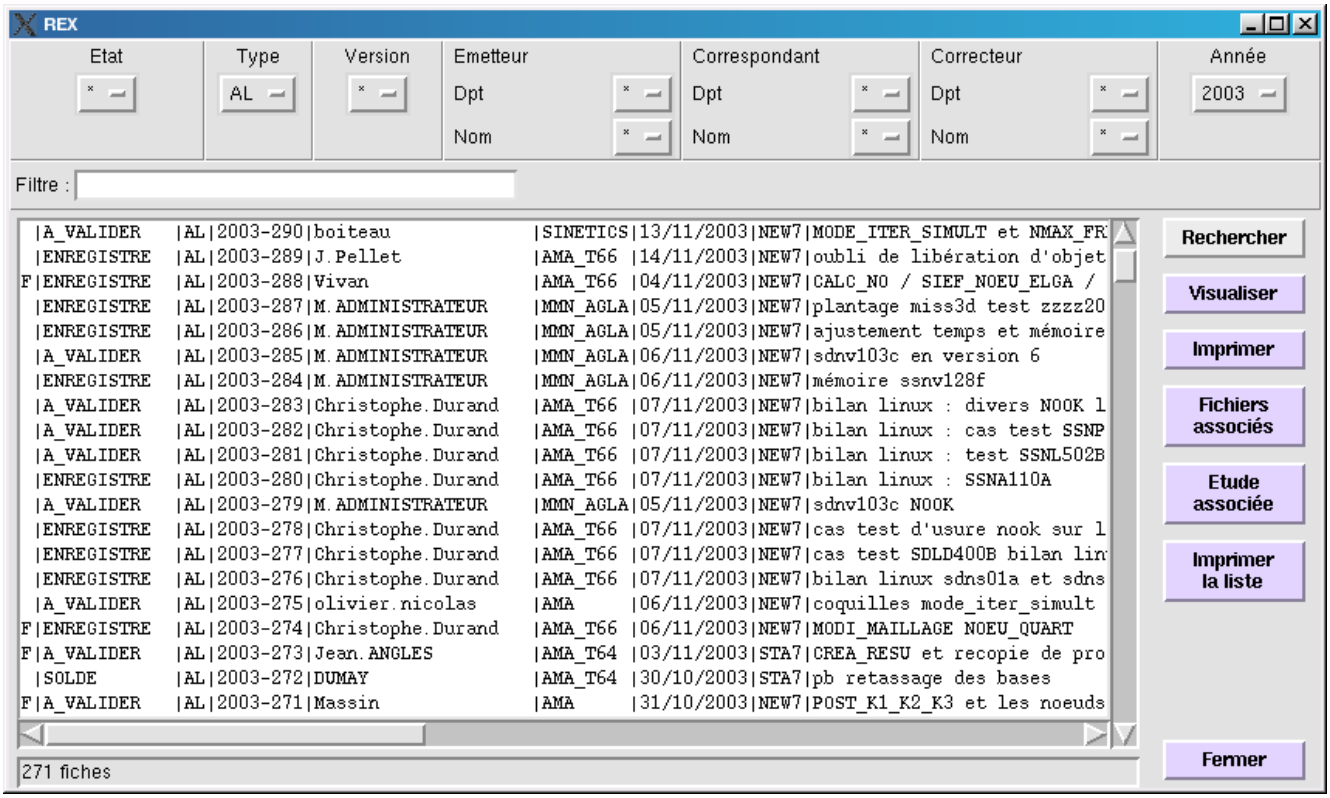


Figure 2.2.3-b : Consultation des fiches

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

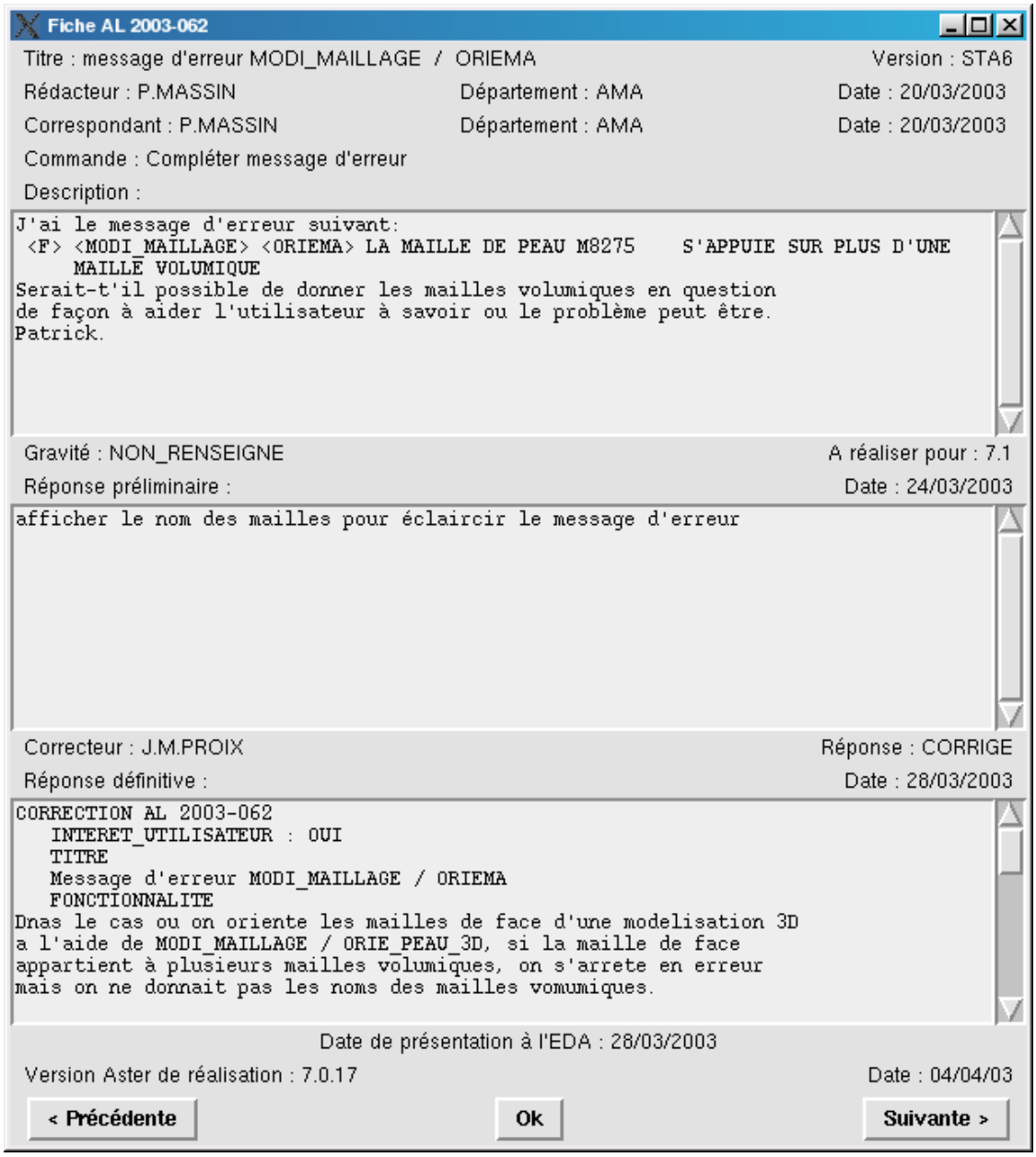

Fiche AL 2003-062
Titre : message d'erreur MODI_MAILLAGE / ORIEMA
Version : STA6
Rédacteur : P.MASSIN
Département : AMA
Date : 20/03/2003
Correspondant : P.MASSIN
Département : AMA
Date : 20/03/2003
Commande : Compléter message d'erreur
Description :
J'ai le message d'erreur suivant:
<F> <MODI_MAILLAGE> <ORIEMA> LA MAILLE DE PEAU M8275 S'APPUIE SUR PLUS D'UNE MAILLE VOLUMIQUE
Serait-t'il possible de donner les mailles volumiques en question
de façon à aider l'utilisateur à savoir ou le problème peut être.
Patrick.
Gravité : NON_RENSEIGNE
A réaliser pour : 7.1
Réponse préliminaire :
Date : 24/03/2003
afficher le nom des mailles pour éclaircir le message d'erreur
Correcteur : J.M.PROIX
Réponse : CORRIGE
Réponse définitive :
Date : 28/03/2003
CORRECTION AL 2003-062
INTERET_UTILISATEUR : OUI
TITRE
Message d'erreur MODI_MAILLAGE / ORIEMA
FONCTIONNALITE
Dnas le cas ou on oriente les mailles de face d'une modelisation 3D
a l'aide de MODI_MAILLAGE / ORIE_PEAU_3D, si la maille de face
appartient à plusieurs mailles volumiques, on s'arrete en erreur
mais on ne donnait pas les noms des mailles vomumiques.
Date de présentation à l'EDA : 28/03/2003
Version Aster de réalisation : 7.0.17
Date : 04/04/03
< Précédente
Ok
Suivante >

Figure 2.2.3-c : Visualisation d'une fiche

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.3 Paramètres de soumission

Les paramètres du calcul sont fournis dans la partie droite de la fenêtre principale.

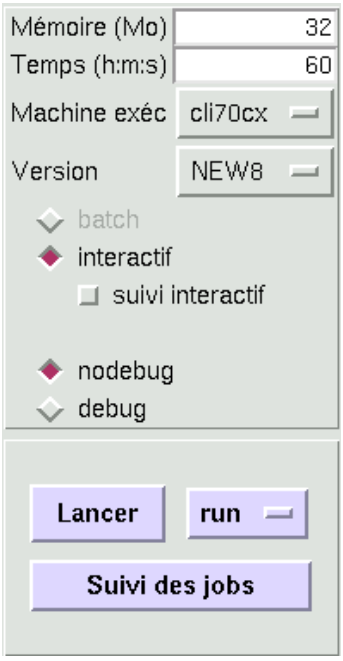


Figure 2.3-a : Zone des paramètres du calcul

On définit la quantité de mémoire utilisée pour le job (en mégaoctets), le temps maximum du calcul (en secondes, minutes:secondes ou heure:minutes:secondes).
On choisit sur quelle machine le calcul est exécuté, la version de *Code_Aster* utilisée, si le calcul est soumis en batch ou en interactif.
debug/nodebug : pour une étude sans surcharge, on précise quel exécutable on souhaite utiliser (sous réserve que les deux soient disponibles) ; lors d'une surcharge, on choisit de compiler avec ou sans les informations de debug.

Le bouton « Lancer » exécute les actions en fonction des onglets cochés.
Le bouton « Suivi des jobs » ouvre la fenêtre décrite après.

Mode de fonctionnement (pour une ETUDE) :

Lors du lancement d'une étude (avec ou sans surcharge), un bouton d'option est disponible à côté du bouton « Lancer ». Trois modes de lancement sont disponibles :

- « run » : exécute l'étude (fonctionnement classique),
- « dbg » : lance l'étude en utilisant le debugger,
- « pre » : prépare le répertoire de travail sans exécuter l'étude.

Quand on sélectionne « dbg » ou « pre », le mode « debug » est choisi par défaut.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

2.4 Barre d'état

La zone de texte située tout en bas de la fenêtre principale fournit de l'aide lorsque l'on navigue dans les menus, ou bien quand le pointeur passe au dessus des boutons de l'interface.
Lors du lancement d'un calcul, les opérations en cours sont affichées ici.

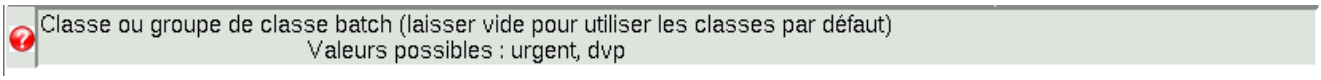


Figure 2.4-a : Barre d'état

2.5 Suivi des jobs

Cette fenêtre fournit des informations sur l'état des calculs lancés, qui apparaissent sous forme d'une liste.
« Editer » se décompose en « Fichier output » et « Fichier error » qui donne accès aux messages envoyés par le job sur la sortie standard (`stdout`) et la sortie d'erreur (`stderr`). Un double-clic sur un job de la liste affiche également le fichier output.
« Supprimer » efface les jobs sélectionnés de la liste, les fichiers liés à ce job dans le flasheur et interrompt le calcul si celui-ci n'est pas terminé.

« Actualiser » interroge les serveurs sur lesquels des calculs sont en cours. La case à cocher permet de réactualiser automatiquement à une fréquence définie dans *Configuration/Interface*.

« Rechercher » permet de consulter les dernières lignes du fichier message d'un job *en cours d'exécution* (il ne fait rien sur un job terminé). On peut utiliser la zone de texte « Filtre » pour n'afficher que les lignes contenant la chaîne de caractères indiquée.

Nouveauté

On retrouve les mêmes fonctions dans le menu contextuel qui apparaît en cliquant avec le bouton droit sur un ou plusieurs jobs. On peut ainsi n'actualiser qu'un seul job parmi plusieurs non terminés.

Chaque ligne correspond à un job, on trouve 13 colonnes :

- Le numéro du job (en batch), numéro du processus en interactif
- Le nom du job (nom du profil pour une étude, une surcharge, ou nom de la fonction AGLA)
- Date de soumission
- Heure de soumission
- Etat du job (PEND, RUN, SUSPENDED, ENDED)
- Diagnostic du job (OK, NOOK, <A>_ALARME, <F>_ERREUR, <F>_ARRET_ANORMAL…)
- Nom de la queue en batch ou « interactif »
- Temps CPU de l'exécution *Aster*
- Login sur le serveur de calcul utilisé
- Adresse du serveur de calcul utilisé
- Machine de calcul (nom du nœud pour un cluster)
- Version d'astk
- Indicateur batch/interactif

Le diagnostic émis par `as_run` apparaît en anglais, car il est prévu de les traduire globalement, mais cela n'a pas encore été réalisé.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3 Boite de sélection de fichiers : bsf

bsf est un outil livré avec astk qui peut être lancé seul. Il s'agit d'un explorateur de fichiers qui permet de naviguer sur la machine locale, comme un explorateur de fichiers classique, et aussi sur les différents serveurs distants configurés.

bsf utilise la configuration des serveurs de astk, notamment les champs nécessaires à la connexion (adresse IP, login) et les commandes pour ouvrir un terminal ou un éditeur.
La lecture de la configuration n'est faite qu'au démarrage de bsf, si on modifie la configuration dans astk, il faut donc fermer la bsf puis l'ouvrir de nouveau.

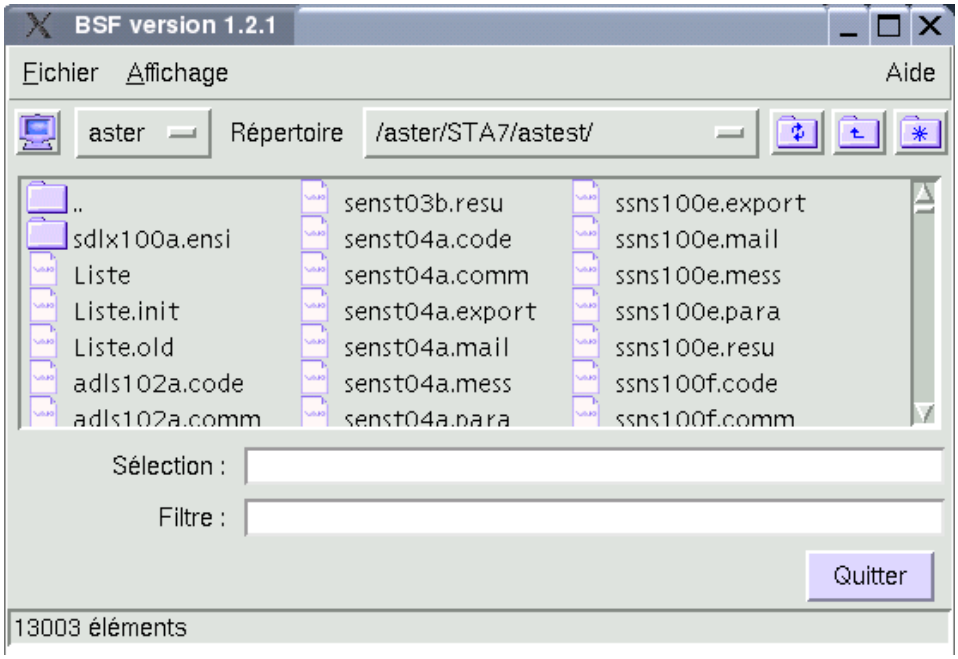


Figure 3-a : bsf

Remarque :

> *bsf ne traite pas les noms de fichiers et répertoires contenant les espaces (il en résulte un décalage dans les noms et les types des objets suivants).*

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.1 Navigation

On trouve deux listes déroulantes dans la fenêtre de la bsf.
La première permet de passer d'un serveur à un autre, la deuxième mémorise la liste des douze derniers répertoires où l'utilisateur a effectué une action (édition, copie…). La première entrée de cette liste est '----- Direct -----', qui permet d'aller directement dans un répertoire sans parcourir l'arborescence.
D'une manière générale, lorsque le répertoire demandé n'existe pas, on retourne dans le HOME défini pour le serveur courant.
On peut configurer cette liste et fixer certains répertoires (pour qu'ils restent présents dans la liste) en cliquant avec le bouton droit sur celle-ci (cf. Figure 3.1-a).

- : rafraîchit le contenu du répertoire courant ;
- : remonte au répertoire parent ;
- : propose de créer un nouveau répertoire dans le répertoire courant (et se place dans ce nouveau répertoire) ;
- : ouvre un terminal sur le serveur actuel.

La barre d'état donne des indications sur la signification de ces boutons lorsque le pointeur de la souris passe au dessus.

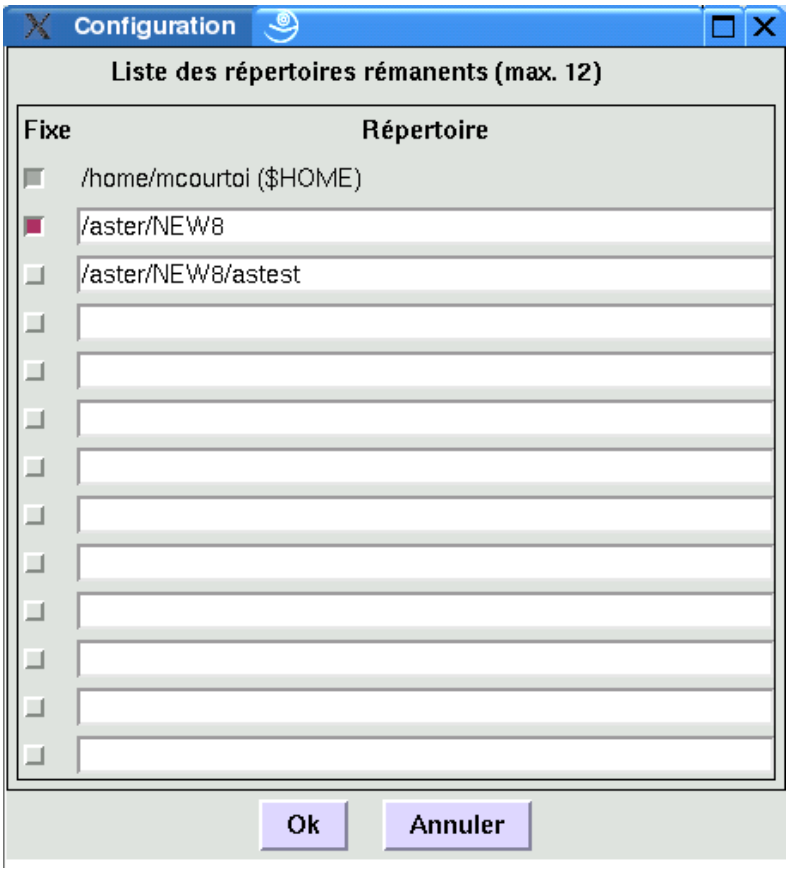


Figure 3.1-a : Fenêtre de configuration des répertoires rémanents

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

3.2 Menu Fichier et menu contextuel

On retrouve certaines entrées semblables au menu contextuel des fichiers dans les onglets (voir [§2.2.1]) : Ouvrir, Editer, Imprimer, Ouvrir avec…, Propriétés.

« Copier » : permet de copier un ou plusieurs fichiers/répertoires (raccourci par CTRL+C).
« Couper » : idem Copier, sauf que les fichiers originaux seront supprimés (raccourci par CTRL-X).
« Coller » : réalise effectivement la copie ou le déplacement des fichiers (raccourci par CTRL+V).
« Supprimer » : efface les fichiers/répertoires sélectionnés.

« Nouveau fichier/Nouveau répertoire » : permettent respectivement de créer un fichier ou un répertoire dans le répertoire courant.

« Exécuter une commande » : donne la possibilité d'exécuter une commande dans le répertoire courant, la sélection courante est proposée sur la ligne de commande, il faut choisir un des shells disponibles sur le serveur.

3.3 Menu Affichage

On peut choisir d'afficher ou non les fichiers dits cachés (commençant par '.') en cochant la case prévue.
bsf affiche les répertoires puis les fichiers, on peut les trier par ordre alphabétique du nom ou en fonction de la date dans l'ordre normal ou inverse en modifiant les options du menu.

Les préférences d'affichage sont conservées si l'on sélectionne « Enregistrer les préférences ».

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 Configuration

La première fois que l'utilisateur lance l'interface, il dispose d'une configuration par défaut qui a été définie lors de l'installation. La configuration est ensuite stockée dans le répertoire `$HOME/.astkrc`. Si l'utilisateur veut revenir à la configuration d'origine, il doit effacer ce répertoire et relancer astk.

Remarque n°1

A chaque fois que l'on définit une commande à exécuter (terminal, éditeur…), il est conseillé d'indiquer le chemin absolu (depuis la racine) pour éviter que la commande ne soit pas trouvée si la variable `$PATH` est incorrecte.

Remarque n°2

En général, astk complète les commandes qui ont besoin d'afficher des fenêtres avec l'argument ' `-display ECRAN` ', où `ECRAN` est la valeur courante de la variable `$DISPLAY` . Si la commande utilisée n'accepte pas cet argument, il suffit de préciser la bonne syntaxe avec le code @D qui sera remplacé par la variable courante de `$DISPLAY` (par exemple : `mon_editeur --display=@D` , cf. Menu Outils).
On peut vérifier la valeur courante de `$DISPLAY` , menu Aide/Fenêtre des messages : astk indique dans les premières lignes par exemple :

```
<INFO> Affichage des applications sur mach00.domain.org:0.0
```

C'est cette valeur qui sera fournie aux machines distantes censées afficher une fenêtre sur la machine cliente.

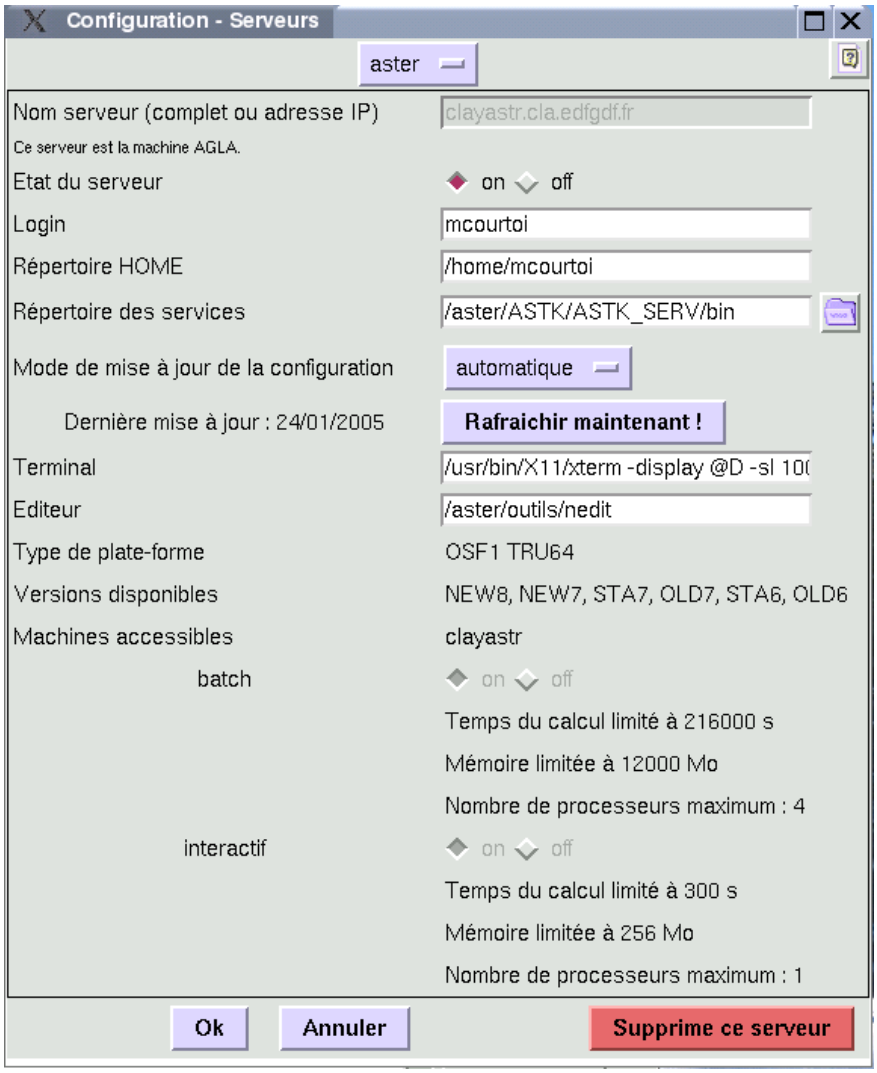
La partie serveur d'astk (ASTK_SERV) inclus des outils simplifiant certaines tâches des développeurs comme la consultation du code source ou la mise à jour d'une version locale de développement.
Le fichier `$HOME/.astkrc/config` contient donc deux informations nécessaires pour contacter la machine de référence : le nom complet de ce serveur et le login de l'utilisateur. Ce dernier doit être défini dans ce fichier pour éviter l'alarme suivante :

```
<A>_ALARM          remote connection may fail :
devel_server_user not defined in /home/xxxxxx/.astkrc/config
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4.1 Serveurs

On accède à la fenêtre de configuration par le menu *Configuration/Serveurs*.



Configuration/Serveurs

Le premier bouton permet de passer d'un serveur à un autre et d'ajouter un « Nouveau serveur ».
Les champs sont :

- Nom complet ou adresse IP : il s'agit du nom du serveur sur le réseau ; on peut indiquer son nom complet avec le nom de domaine (par exemple : linux.labo.univ.fr) ou son adresse IP (par exemple : 156.98.254.36).
- Etat du serveur : on peut mettre sur « off » un serveur temporairement inaccessible.
- Login : identifiant avec lequel on se connecte au serveur.
- Répertoire HOME : répertoire par défaut lorsque l'on arrive sur ce serveur avec la bsf.
- Répertoire des services : répertoire où sont installés les services sur ce serveur (indiquer le chemin complet, par exemple : `/aster/ASTK/ASTK_SERV/bin`), laisser vide pour un serveur de fichiers.
- Mode de téléchargement de la configuration : aucun (pour un serveur de fichiers), manuel (il faut cliquer sur le bouton « Télécharger maintenant » pour récupérer la configuration Aster du serveur), automatique (astk interroge le serveur au démarrage tous les 30 jours).
- Dernier téléchargement : date de la dernière mise à jour des informations de configuration.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

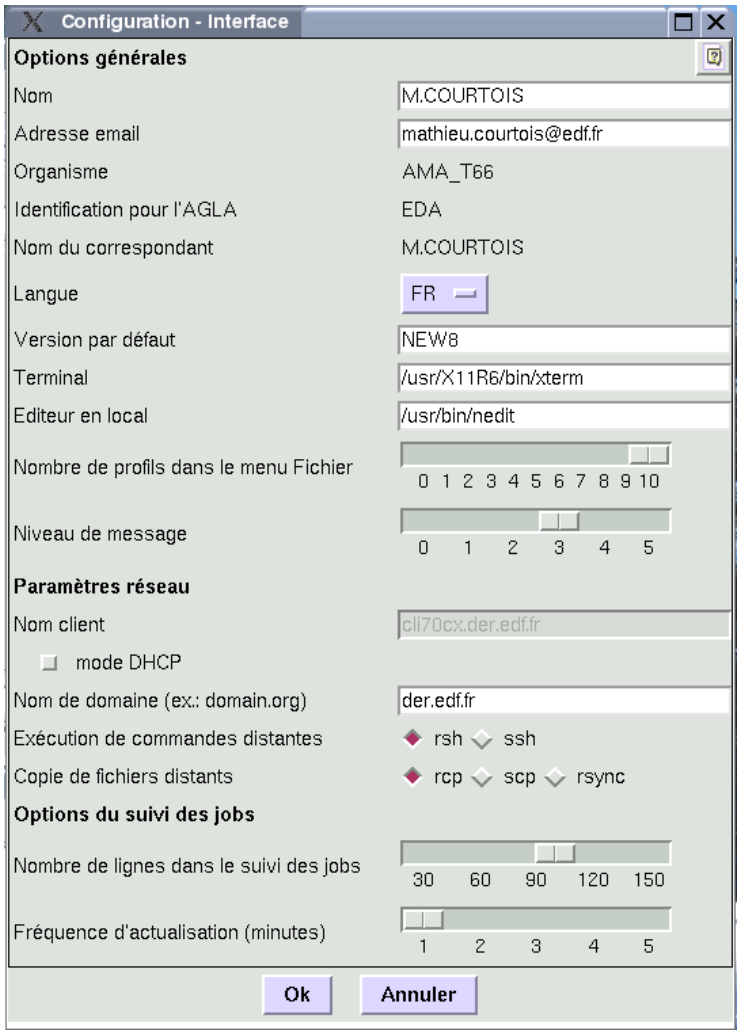
- Terminal : commande pour ouvrir un terminal sur le serveur. Ceci permet d'ouvrir une fenêtre de commandes sur le serveur quand on utilise la bsf, et de suivre interactivement un calcul (le terminal doit accepter l'option `-e` pour exécuter une commande, `xterm` convient).
- Editeur : éditeur texte (par exemple, `nedit`). La procédure d'installation choisit un éditeur parmi (et dans cet ordre) : `nedit`, `xemacs`, `emacs`, `xedit`, `vi`.

Les valeurs suivantes sont retournées par le service `as_info` (rien pour un serveur de fichiers) et dépendent donc de la configuration de la partie serveur d'astk :

- Type de plate-forme.
- Versions disponibles
- Machines accessibles : liste des nœuds de calcul accessibles depuis ce serveur.
- Batch/interactif : précise si le serveur accepte le lancement en batch, en interactif et fournit les limites en mémoire, temps CPU, nombre de processeurs fixés sur le serveur.
  Seul LSF est supporté comme logiciel de gestion de batch, le support de PBS est encore incomplet et expérimental.

4.2 Interface

On accède à la fenêtre de configuration par le menu *Configuration/Interface*.



Configuration/Interface

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Cette fenêtre permet de renseigner les informations personnelles de l'utilisateur, de choisir la langue utilisée par l'interface.
Pour ceux qui ont accès à la machine de référence, l'instance AGLA est affichée (EDA pour développeur, UTL pour utilisateur…). Pour les EDA, l'organisme et le nom du correspondant est automatiquement rempli. Les utilisateurs doivent le faire eux-mêmes.

Ensuite, on trouve la version qui sera sélectionnée par défaut, les commandes pour accéder à un terminal et un éditeur (comme pour les serveurs).
« Nombre de profils dans le menu Fichier » permet de conserver le nom des N derniers profils ouverts de manière à les rappeler rapidement.

« Niveau de message » indique le niveau de détails des messages écrits dans la fenêtre des messages du menu *Aide*. Niveau=0 : seuls les messages `<INFO>` et `<ERREUR>` sont écrits ; les niveaux supérieurs permettent de debugger le comportement de l'interface. Le niveau 1 est conseillé, il permet de voir les messages d'erreur pouvant apparaître lors des problèmes de communication avec les serveurs distants.

On doit ensuite préciser le nom de domaine réseau de la machine. Par exemple, `domain.org` si le nom complet de la machine est `mach00.domain.org`. Si le nom de domaine est laissé vide, un message d'alarme est affiché au démarrage car les machines distantes risquent de ne pas pouvoir contacter la machine locale `mach00` avec son nom court. Le nom de la machine tel qu'il sera utilisé par les machines distantes est indiqué dans le champ Nom Client. Si le mode DHCP est actif, c'est dans ce champ que l'on peut indiquer l'adresse IP de la machine locale.

Pour le suivi des jobs, on peut choisir le nombre de lignes affichées lorsque l'on visualise le fichier output en cours de job (bouton « Rechercher »), et la fréquence d'actualisation de la liste.

4.3 Outils

On sélectionne l'outil à configurer avec la liste déroulante, ou bien on ajoute un nouvel outil.
Des outils standards sont pré-définis (les minuscules/majuscules sont prises en compte dans les noms d'outils). *Les outils sont nécessairement exécutés sur la machine « Local »* (où est lancée l'interface).

En général, les outils sont lancés soit sur un fichier d'un onglet (ETUDE, TESTS ou SURCHARGE), soit sur un fichier lorsque l'on parcourt le système de fichiers avec la bsf.

On définit simplement la ligne de commande nécessaire au lancement d'un outil (chemin absolu conseillé), on peut placer les codes @F, @R, @f, @D dans la ligne de commande (voir [§ 2.1.3]) pour passer correctement un fichier à l'outil. On peut mettre ces codes entre parenthèses pour pouvoir lancer l'outil seul, sans fichier en argument.

Des types de fichiers peuvent être associés à l'outil. Le type pris en compte pour lancer l'outil est l'extension lorsque l'on parcourt les fichiers avec la bsf, le type de la liste déroulante quand il s'agit d'un onglet.

On peut choisir si l'outil est utilisable sur un fichier distant. Dans ce cas, astk se charge de ramener le fichier en question sur la machine locale dans un répertoire temporaire, de lancer l'outil, puis de redéposer le fichier sur le serveur distant (même s'il n'a pas été modifié par l'outil).

4.4 Impression

Les commandes d'impression sont définies par le menu *Fichier/Impression*.
Sur le même principe que les outils, on définit n'importe quelle commande d'impression (lp, lpr, a2ps…) qui contient le code correspondant au nom du fichier (@F, voir [§ 2.1.3]). @P est remplacé par le nom de la file d'impression.

Voici une liste de commande classique utilisant GNU a2ps :

Format 2 colonnes/page :

```
a2ps -P @P -2 -l82 --footer --print-anyway=yes --sides=tumble @F
```

Format 136 caractères/ligne :

```
a2ps -P @P -r -l136 --columns=1 --footer --print-anyway=yes
--sides=tumble @F
```

Format "transparent" :

```
a2ps -P @P -r --columns=1 --print-anyway=yes --sides=simplex @F
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5 Comment faire une étude ?

Dans ce paragraphe, on décrit étape par étape comment utiliser astk pour réaliser une étude.

L'étude consiste à calculer la réponse en flexion d'une tuyauterie coudée. On dispose des éléments suivants :

- Un fichier de commande *Aster* : `demo01a.comm`
- La description de la géométrie réalisée avec Gmsh : `demo01.geo`
- Le maillage construit par Gmsh : `demo01a.msh`

On produit les résultats suivants :

- Un fichier de maillage *Aster* : `demo01a.mail`
- Un fichier résultat au format Gmsh (champs de déplacements, contraintes…) : `demo01a.pos`
- Les fichiers classiques de message et résultat Aster : `demo01a.mess` et `demo01a.resu`

Dans l'exemple, on place tous les fichiers dans le répertoire `/home/tutorial/demo01`.

Remarque :

> *Dans le cas d'une étude avec plusieurs fichiers de commandes, tous les fichiers doivent être de type « comm », associés à l'unité logique 1 et c'est l'extension qui détermine l'ordre d'exécution : .comm, puis .com0, .com1, …, .com9 (il peut y avoir des trous).*

5.1 Création du profil

On lance l'interface qui s'ouvre sur un profil vierge, ou bien si astk est déjà lancé, on choisit *Fichier/Nouveau* dans le menu pour créer un nouveau profil vide.
On se place dans l'onglet ETUDE.

5.2 Sélection des fichiers

5.2.1 Définition d'un chemin de base

Dans l'onglet ETUDE, on choisit un chemin de base pour simplifier l'accès aux fichiers.

On clique sur l'icône , on choisit le répertoire `/home/tutorial/demo01`.

5.2.2 Ajout de fichiers existants

On ajoute le fichier de commandes en cliquant sur , la sélection de fichier s'ouvre directement dans le chemin de base que l'on vient de définir. Il ne reste qu'à sélectionner le fichier `demo01a.comm` (double-clic ou simple clic + ok), et le fichier apparaît dans la liste. Notons qu'astk identifie le type de ce fichier à partir de son extension « comm », le numéro d'unité logique est positionné à 1, la case « D » (donnée) est cochée.

On fait de même pour le fichier de maillage au format Gmsh (`demo01a.msh`). astk reconnaît l'extension « msh », le numéro d'unité logique est positionné à 19, la case « D » est cochée.

5.2.3 Ajout de fichiers…

Sauf si une exécution a déjà eu lieu, les fichiers résultats n'existent pas encore, on ne peut donc pas les ajouter en parcourant l'arborescence.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5.2.3.1 …en insérant une ligne vide

Le maillage au format Gmsh sera relu et convertit dans le fichier de commandes *Aster* par la commande PRE_GMSH en maillage au format *Aster*. On peut récupérer ce maillage en ajoutant un fichier de type « mail » sur l'unité logique 20.

On clique sur , une ligne est ajoutée dans la liste. On choisit le type « mail » dans la liste (ce qui a pour effet de positionner le numéro d'unité logique à 20). On indique le nom /home/tutorial/demo01/demo01a.mail ou demo01a.mail ou ./demo01a.mail (puisque l'on peut indiquer le nom du fichier en relatif par rapport au chemin de base). Le fichier est produit par l'exécution, on coche donc la case « R » (résultat) et on décoche « D ».

Remarque

La commande PRE_GMSH utilise par défaut les numéros 19 et 20 avec les entrées/sorties, si l'on modifie le fichier de commande pour relire ou écrire les fichiers de maillage sur d'autres unités, il faut être cohérent avec les numéros indiqués dans astk.

5.2.3.2 …avec « Valeur par défaut »

On pourrait continuer ainsi pour ajouter les autres fichiers, mais on va utiliser la fonction « Valeur par défaut » pour les fichiers suivants. Cette fonction utilise le nom du profil astk pour construire les valeurs par défaut (voir [*§ 2.2.1]/Menu contextuel*), on va donc enregistrer le profil.
On choisit *Enregistrer sous…* dans le menu *Fichier*, on va avec le navigateur dans le répertoire /home/tutorial/demo01, et dans la ligne *Sélection*, on tape demo01a (l'extension .astk est automatiquement ajoutée).
Notons que le titre de la fenêtre principale d'astk donne le nom du profil courant. Le titre est maintenant : ASTK version 1.2.1 - demo01a.astk - /home/tutorial/demo01

On insère une ligne vide en cliquant sur , on choisit le type de fichier « pos », puis on clique avec le bouton droit dans la case du nom de fichier et on choisit « Valeur par défaut » : astk construit un nom de fichier à partir du nom de profil (en retirant l'extension) et du type « pos », soit /home/tutorial/demo01a/demo01a.pos. De plus, astk essaie de substituer le chemin de base pour plus de lisibilité, on voit ainsi : ./demo01a.pos.
La case « R » a été cochée, et le numéro d'unité logique fixé à 37. Supposons que dans le fichier de commande, on ait indiqué :

```
IMPR_RESU(UNITE=30, ...)
```

on modifie donc le numéro d'unité logique en conséquence, il suffit de cliquer sur l'ancienne valeur, de l'effacer et de taper 30. Seuls deux chiffres sont affichés dans cette case, pour éviter les erreurs, astk vérifie que les numéros d'unité logique sont compris entre 1 et 99.

De même, on ajoute un fichier de type « mess » et un de type « resu » de cette manière (laisser les numéros d'unité logique par défaut).

5.2.4 Supprimer un fichier

Pour supprimer une ligne de la liste des fichiers, il suffit de la sélectionner en cliquant dans la zone où l'on indique le nom du fichier et de cliquer sur l'icône .

Remarque :

Seule la référence à ce fichier dans le profil astk est oubliée, le fichier lui-même n'est pas effacé !

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5.3 Lancement du calcul

Les fichiers données et résultats sont sélectionnés, on ajuste les paramètres du calcul (voir [§ 2.3]), et on clique sur le bouton « Lancer ».
On prend soin de cocher la case qui se trouve juste à côté de ETUDE pour signaler que l'on souhaite utiliser le contenu de cet onglet… sinon l'interface nous répond « Rien à lancer ! ».
Si le profil n'a pas encore été enregistré, l'interface demande de choisir un endroit et un nom pour ce profil (voir [§ 5.2.3.2]).

astk appelle le service qui exécute le calcul, et transmet au Suivi des jobs (asjob) le numéro du job (numéro du processus en interactif) et d'autres informations qui vont permettre de suivre l'avancement du calcul. L'état initial du calcul est `PEND` (en attente), quand le calcul commence, il devient `RUN`, puis `ENDED` quand il est terminé (d'autres états sont possibles en batch). Le bouton « Actualiser » appelle le service qui rafraîchit l'état des calculs en cours.

Quand le calcul est terminé, on peut consulter l'output du job en double-cliquant sur le job, ou par *Editer/Fichier output*.

5.4 Consultation des résultats

On peut consulter les fichiers résultats simplement en double-cliquant sur leur nom, ce qui ouvre un éditeur de texte pour les fichiers « mess » et « resu » ; sur le fichier de résultat au format Gmsh, « pos », cela a pour effet d'ouvrir directement ce fichier dans Gmsh. On visualise ainsi la déformée et les isovaleurs (sous réserve que Gmsh ait été installé, et que « pos » soit dans les types de fichiers associés à Gmsh, voir [*§ 4.3]*).

5.5 Utilisation des outils

On peut aussi utiliser astk et le fait que l'on puisse y définir librement des outils pour rassembler dans un profil tous les fichiers nécessaires à une étude même si ceux-ci ne sont pas directement utilisés par *Code_Aster*.

Dans cet exemple, `demo01a.geo` est un fichier que *Code_Aster* ne sait pas relire ; il contient la description de la géométrie, Gmsh l'utilise pour créer le maillage (`.msh`).

On peut néanmoins l'insérer dans le profil (bouton), lui affecter un type quelconque (« libr » par exemple) puisqu'il ne sera pas utilisé lors de l'exécution (cases D, R non cochées).
On peut ouvrir directement la géométrie en faisant *Ouvrir avec…/Gmsh* (clic droit sur le nom du fichier), modifier la géométrie ou les paramètres du maillage, remailler et enregistrer le maillage.

On peut ensuite relancer le calcul sur le nouveau fichier `demo01a.msh`.

Bien évidemment, ceci n'est pas limité à Gmsh ; on peut utiliser d'autres outils (mailleurs, outil de post-traitement, traceur de courbes…) directement depuis astk et accéder ainsi à tous les fichiers d'une étude depuis un profil avec l'outil adéquat.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

6 Comment réaliser une surcharge ?

On considère dans ce paragraphe que l'utilisateur est familier avec la manipulation des listes de fichiers dans l'onglet ETUDE.

Une surcharge consiste à ajouter ou modifier une partie de *Code_Aster* et à l'utiliser pour réaliser une étude. L'objectif d'une surcharge est de produire un exécutable, un catalogue de commandes compilé et/ou un catalogue d'éléments. On peut aussi surcharger les méthodes python définies par *Aster*, dans ce cas, les sources sont recopiés dans le répertoire d'exécution (il n'y a pas d'objet réceptacle des fichiers python surchargés).
On se place dans l'onglet SURCHARGE.

6.1 Ajout des sources

On procède comme pour ajouter des fichiers pour une ETUDE. On peut sélectionner soit un fichier, soit un répertoire. Il est souvent plus clair et plus pratique de placer ses fichiers sources dans des répertoires.
Si on ajoute un répertoire dans la liste des sources (partie supérieure de l'onglet SURCHARGE) de type « f » (fichiers fortran), tous les fichiers dont l'extension est `.f` seront compilés et utilisés pour faire un nouvel exécutable.

6.2 Définir les résultats de la surcharge

Les fichiers C (type « c ») et fortran (type « f ») permettent de construire un exécutable : type « exec ».
Les catalogues de commandes « capy » permettent de construire un catalogue de commandes compilé : type « cmde » (répertoire contenant les fichiers `cata.py` et `cata.pyc`).
Les catalogues d'éléments, d'options et de grandeurs « cata » servent à produire un catalogue d'élements compilés : « ele ».

6.3 Prise en compte de la surcharge

Pour que les données renseignées dans l'onglet SURCHARGE soient prises en compte, il faut cocher la case située juste à droite du bouton SURCHARGE (l'onglet SURCHARGE est toujours coché dans ce paragraphe).

Il faut absolument un réceptacle en résultat correspondant aux sources en données (« D » coché).
S'il y a des répertoires « c » et/ou « f » en « D »onnée , il faut un « exec » en « R »ésultat ; de même pour « capy » avec « cmde » et « cata » avec « ele ».

On peut préparer la surcharge indépendamment de l'étude (conseillé) ou bien faire la surcharge et lancer l'étude dans la foulée.

6.3.1 Préparer la surcharge seule, puis lancer une étude

Pour préparer la surcharge seule :
- décocher l'onglet ETUDE
- mettre les sources en « D »onnée
- mettre les réceptacles correspondants en « R »ésultat seulement
- « Lancer ».

La surcharge construit les résultats (exécutable, catalogues…) à partir des données (les sources).

Lancement de l'étude en utilisant cette surcharge :
- cocher l'onglet ETUDE
- décocher l'indicateur « D » pour les sources
- mettre les réceptables en « D »onnée seulement
- « Lancer ».

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Les produits de la première étape sont alors utilisés comme données pour lancer l'étude.
Attention : Ne pas décocher SURCHARGE, sinon l'étude sera lancée avec la version standard non surchargée.

Il est rare que la compilation réussisse au premier essai, en séparant les deux tâches, on voit immédiatement si la compilation se passe mal, ou si l'exécutable n'est pas produit.

6.3.2 Faire la surcharge et lancer l'étude en même temps

Pour enchaîner les deux étapes automatiquement :
- cocher l'onglet ETUDE
- mettre les sources en « D »onnée
- mettre les réceptacles correspondants en « D »onnée et « R »ésultat
- « Lancer ».

En cas de problème de compilation, il se peut qu'un exécutable soit produit sans prendre en compte une partie des sources en donnée. Il faut donc regarder attentivement l'output de la compilation. Pour cela, la démarche du paragraphe précédent est conseillée.

7 Comment lancer une liste de tests ?

Il est nécessaire d'aborder ce point après les deux précédents. En effet, lancer une liste de tests n'a d'intérêt que pour valider une surcharge soit vis à vis de la non régression des fonctionnalités originales du code, soit sur un ensemble de cas testant une nouvelle fonctionnalité.

Pour utiliser cette possibilité, il faut cocher la case située juste à droite du bouton TESTS (ce qui décoche automatiquement ETUDE). Dans la plupart des cas, on utilise une version surchargée, donc dans ce cas l'onglet SURCHARGE est également coché.

Remarque n°1

> *Il faut absolument préparer la surcharge indépendamment du lancement de test comme au § « Préparer la surcharge seule, puis lancer une étude ». On prépare la surcharge, puis on lance les tests avec les résultats de cette surcharge.*

Remarque n°2

> *Sur la machine de référence, le lancement d'une liste de tests doit être fait uniquement en batch.*

Les données sont très simples, on fournit un fichier de type « list » (voir [*§ 1*]) ; « rep_test » indique où se trouvent les fichiers nécessaires au lancement des tests (« données » des tests : `.comm`, `.mail`...) ; « resu_test » permet d'écrire les résultats des tests dans un autre répertoire.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

8 Questions fréquentes

« Rien ne se passe quand on essaie de lancer un calcul, d'éditer un fichier ou d'ouvrir un terminal sur un serveur distant » *ou bien* ***« Code retour = 2, Profile copy failed dans la fenêtre des messages au lancement d'un calcul »***
Il est probable que la machine locale ne puisse pas communiquer correctement avec le serveur distant. Les communications utilisant le protocole rcp/rsh, les fichiers `.rhosts` sur la machine locale et sur les serveurs distants doivent être correctement renseignés. Si on utilise le protocole scp/ssh, il faut que les clés privées et publiques sont cohérentes.
On peut le vérifier en quittant astk, et en le relançant de cette manière :

```
astk --debug 0 --check
```
(`--debug 0` sert à limiter les impressions)

Lire attentivement les informations qui sont imprimées. astk fournit des informations sur la variable d'environnement DISPLAY (vérifier qu'elle est correctement définie), et teste la connexion entre la machine où est lancée astk et les serveurs distants. En cas de problème de connexion avec un serveur, astk suggère une modification du fichier `.rhosts` de la machine cible.

« Quand on soumet un calcul en batch, on a le message : Le numéro du job et la classe n'ont pas pu être récupérés »

Le temps ou la mémoire demandée est probablement au delà des limites des classes de travaux. Voir dans le fenêtre des messages, il y a probablement un message du gestionnaire de batch du style "Cannot exceed queue's hard limit(s)".
Diminuer le temps ou la mémoire, ou choisir explicitiment une classe batch qui convient dans les options supplémentaires.

« En cliquant sur Valeur par défaut le nom de fichier reste vide »
Cette fonction utilise le nom du profil pour déterminer une valeur par défaut, il faut enregistrer le profil avant de pouvoir utiliser cette fonction.

« Sur la machine de référence : les comportements en batch et en interactif sont différents »
C'est possible. En batch, *Code_Aster* est lancé par les scripts de l'AGLA, en interactif, on s'appuie sur `as_run`. Signaler dans une « AO » les différences.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

Annexe 1Utilisation du service as_run (de ASTK_SERV)

La lecture de ce chapitre est réservé aux utilisateurs qui souhaitent lancer *Code_Aster* « à la main », sans utiliser l'interface et à ceux qui ont installé et maintiennent une version locale.

Lorsque l'on utilise l'interface astk pour lancer des calculs, celle-ci en tant que client fait appel à des services proposés par un serveur qui peut se trouver sur la même machine ou une machine distante (dans le cas où le serveur est distant, il y a des échanges de fichiers et une commande shell à travers le réseau (protocole rsh ou ssh) que nous ne détaillons pas ici).

Plusieurs services sont appelés par l'interface :

- `as_info` : récupère les informations de configuration du serveur : les versions et machines de calcul disponibles, les limites en interactif et en batch ;
- `as_actu` : récupère l'état d'un calcul en cours ou terminé ;
- `as_tail` : récupère les dernières lignes de l'output du calcul (stdout, sortie standard) ;
- `as_del` : arrête un calcul en cours et/ou supprime les fichiers du flasheur associés à ce calcul ;
- `as_edit` : ouvre le fichier output d'un calcul (fichier .o du flasheur) ou error (fichier .e) dans un éditeur ;
- `as_exec` : lance les actions indiquées dans le profil en argument (fichier de type .export).

`as_run` est un service appelé par `as_exec` dans le cadre d'une exécution via l'interface, il se limite à lancer une exécution de *Code_Aster*, avec ou sans surcharge, une étude simple ou une liste de tests.

Les options de `as_run` sont données en tapant : `as_run --help`, on obtient :

```
usage: as_run action [options] [arguments]

  Functions :

   - Return the main subroutine of a code_aster command :
      as_run --getop [options] commande[.capy]

   - Return current release number of the development version :
      as_run --getversion [options]

   - Show a source file : fortran, c, python, capy, cata, histor or test :
      as_run --show [options] obj1 [obj2...]

   - Copy a source file in current directory :
      as_run --get [options] obj1 [obj2...]

   - Build a code_aster version (executable, libraries, catalogs) :
      as_run --make [--version_dev=VERS]

   - Perform one or several updates of a development version :
      as_run --update [options] fich1.tar.gz [fich2.tar.gz...]

   - Download available updates from a server and apply them to the current
development version :
      as_run --auto_update [options]

   - Execute the execution described by the profile (default action) :
      as_run --run [options] profile


options:
  --version             show program's version number and exit
```


Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
  -h, --help            show this help message and exit
  -v, --verbose         print status messages
  -g, --debug           print debugging information
  -f, --force           force operations which can be cached (download,
                        compilation...)
  --remote_shell_protocol=REMOTE_SHELL_PROTOCOL
                        remote protocol used for shell commands
  --remote_copy_protocol=REMOTE_COPY_PROTOCOL
                        remote protocol used to copy files and directories
  --editor=EDITOR       editor command
  --devel_server_user=DEVEL_SERVER_USER
                        login on the development server  (name/ip address is
                        usually set in ASTK_SERV/conf/config)

  Options for maintenance operations:
    --vers=VERS         Code_Aster version to used
    -a, --all           get all the files of the test
    --version_dev=VERS  development version of Code_Aster to update
    -l, --local         files will not been searched on a server but on the
                        local machine
    --nolocal           force remote files search (reverse of --local)
```

Précisions supplémentaires :

- On peut créer un lien vers `as_run` pour n'importe quelle action pour simplifier l'appel (ceci est surtout utile pour les actions `get`, `show` et `getop` que l'on appelle souvent) :
  ```
  get -> as_run
  ```
  On peut ensuite utiliser `get fichier.f` au lieu de `as_run --get fichier.f`

- Quand on récupère un fichier source avec `get`, `show` ou `getop`, celui-ci est mis dans `/tmp/astk_`login`/cache`. Si on demande de nouveau le même fichier, celui-ci est pris directement dans ce répertoire sauf si l'option `--force` est activée ; les fichiers qui ont plus d'une journée sont automatiquement supprimés du cache.

- `--local`/`--nolocal` : ces deux options permettent de passer outre la valeur définie dans le fichier de configuration. Si le mode par défaut est de chercher les fichiers en local, on peut utiliser `--nolocal` pour forcer la recherche sur une machine distante (c'est notamment utile pour chercher les mises à jour de *Code_Aster* sur le serveur www.code-aster.org).
  Inversement, si on récupère systématiquement les fichiers sur un serveur de développement en réseau, on doit utiliser `--local` pour visualiser des fichiers sources installés sur la machine locale.

- Le fichier de configuration principal est `xxxx/ASTK/ASTK_SERV/conf/config` (où `xxxx` est le répertoire d'installation principal d'Aster), l'utilisateur peut définir certaines valeurs dans `$HOME/.astkrc/config` (où `$HOME` est le répertoire par défaut de l'utilisateur), en particulier le nom d'utilisateur à utiliser sur le serveur distant (`devel_server_user`) pour éviter une alarme à chaque lancement de `as_run`.

- `--version_dev` est utilisé par les actions `make`, `update` et `auto_update`. Alors que `--vers` est utilisé par `get`, `show` et `getop`.

Document diffusé sous licence GNU FDL (http://www.gnu.org/copyleft/fdl.html)